



별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto
is a true copy from the records of the Korean Intellectual
Property Office.

출 원 번 호 : 10-2003-0027755
Application Number

출 원 년 월 일 : 2003년 04월 30일
Date of Application APR 30, 2003

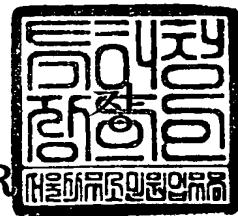
출 원 인 : 주식회사 하이닉스반도체
Applicant(s) Hynix Semiconductor Inc.



2003 년 10 월 24 일

특 허 청

COMMISSIONER



【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0010
【제출일자】	2003.04.30
【발명의 명칭】	데이터 얼라인 마진이 향상된 동기식 메모리 장치
【발명의 영문명칭】	SYNCHRONOUS MEMORY DEVICE FOR ENHANCING DATA ALIGN MARGIN
【출원인】	
【명칭】	주식회사 하이닉스반도체
【출원인코드】	1-1998-004569-8
【대리인】	
【명칭】	특허법인 신성
【대리인코드】	9-2000-100004-8
【지정된변리사】	변리사 정지원, 변리사 원석희, 변리사 박해천
【포괄위임등록번호】	2000-049307-2
【발명자】	
【성명의 국문표기】	윤영진
【성명의 영문표기】	YOOON, Young Jin
【주민등록번호】	720306-1058317
【우편번호】	467-850
【주소】	경기도 이천시 대월면 사동리 현대6차아파트 605-402
【국적】	KR
【심사청구】	청구
【취지】	특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사 를 청구합니다. 대리인 특허법인 신성 (인)
【수수료】	
【기본출원료】	20 면 29,000 원
【가산출원료】	17 면 17,000 원
【우선권주장료】	0 건 0 원
【심사청구료】	8 항 365,000 원
【합계】	411,000 원
【첨부서류】	1. 요약서·명세서(도면)_1통

【요약서】**【요약】**

본 발명은 클럭에 동기되어 입력되는 데이터를 열라인하여 내부회로로 전달하는데 있어서의 데이터 열라인 마진을 증가시켜 고주파에서도 안정적으로 데이터를 입력받아 처리할 수 있는 메모리 장치를 제공하기 위한 것으로, 이를 위해 본 발명은 동작클럭의 라이징에지와 폴링에지에 동기되어 다수의 데이터를 입력받는 동기식 메모리 장치에 있어서, 상기 데이터가 입력되는 타이밍에 클럭킹되는 데이터스트로브 신호를 입력받아, 홀수번째 입력된 데이터스트로브 신호의 라이징에지와 폴링에지를 각각 검출하는 제1 라이징펄스 및 제1 폴링펄스와, 짝수번째 입력된 데이터스트로브 신호의 라이징에지와 폴링에지를 각각 검출하는 제2 라이징펄스와 제2 폴링펄스를 출력하기 위한 데이터스트로브 버퍼링 수단; 상기 제1 폴링펄스에 동기되도록, 상기 동작클럭의 라이징에지에 입력되는 제1 데이터와 상기 동작클럭의 폴링에지에 입력되는 제2 데이터를 열라인시키는 제1 래치수단; 상기 제2 폴링펄스에 동기되도록, 상기 제2 데이터에 이어서 입력되는 제3 데이터와 제4 데이터를 열라인시킨 제3 및 제4 열라인데이터를 출력하는 제3 래치수단; 및 상기 제1 내지 제4 열라인데이터를 짝수데이터 또는 홀수데이터로 선택하여 출력하기 위한 글로벌 입출력라인 드라이버를 구비하는 동기식 메모리 장치를 제공한다.

【대표도】

도 6

【색인어】

반도체, 메모리, 래치, 프리패치, 데이터스트로브, 출력버퍼.

【명세서】

【발명의 명칭】

데이터 얼라인 마진이 향상된 동기식 메모리 장치{SYNCHRONOUS MEMORY DEVICE FOR ENHANCING DATA ALIGN MARGIN}

【도면의 간단한 설명】

도1은 종래기술에 의한 동기식 메모리 장치의 2비트 프리패치 데이터 입력버퍼를 나타내는 블럭구성도.

도2는 도1에 도시된 데이터 입력버퍼의 동작을 나타내는 타이밍도.

도3은 종래기술에 의한 동기식 메모리 장치의 4비트 프리패치 데이터 입력버퍼를 나타내는 블럭구성도.

도4는 도3에 도시된 데이터 입력버퍼의 동작을 나타내는 타이밍도.

도5는 도4에 도시된 데이터 입력버퍼의 동작상의 문제점을 나타내는 타이밍도.

도6은 본 발명의 바람직한 실시예에 따른 동기식 메모리 장치의 4비트 프리패치 데이터 입력버퍼를 나타내는 블럭구성도.

도7은 도6에 도시된 데이터스트로브 버퍼부를 나타내는 블럭구성도.

도8은 도7에 도시된 데이터스트로브 디바이더의 일실시예를 나타내는 회로도.

도9는 도6에 도시된 데이터 입력버퍼의 동작을 나타내는 타이밍도.

도10은 도6에 도시된 데이터 입력버퍼의 데이터 얼라인 마진을 나타내는 타이밍도.

* 도면의 주요부호에 대한 설명

211 ~ 213, 221, 222, 231 ~ 233: 래치

ND1 ~ ND4 : 낸드게이트

NOR1 : 노어게이트

【발명의 상세한 설명】

【발명의 목적】

【발명이 속하는 기술분야 및 그 분야의 종래기술】

<15> 본 발명은 동기식 메모리 장치에 관한 것으로, 특히 데이터를 프리패치(pre-fetch)하여 내부 코어부분으로 전달하는 데이터 입력버퍼부의 데이터 얼라인 마진(align margin)에 관한 것이다.

<16> 반도체 메모리장치는 집적도의 증가와 더불어 그 동작 속도의 향상을 위하여 계속적으로 개선되어 왔다. 동작 속도를 향상시키기 위하여 메모리칩 외부에서 주어지는 클록과 동기되어 동작할 수 있는 소위 동기식(Synchronous) 메모리 장치가 등장되었다.

<17> 처음 제안된 것은 메모리 장치의 외부로부터의 클록의 상승 에지(rising edge)에 동기되어 하나의 데이터 핀에서 클록의 한 주기에 걸쳐 하나의 데이터를 입출력하는 이른바 SDR(single data rate) 동기식 메모리 장치이다.

<18> 그러나 SDR 동기식 메모리 장치 역시 고속 동작을 요구하는 시스템의 속도를 만족하기에는 불충분하며, 이에 따라 하나의 클록 주기에 두 개의 데이터를 처리하는 방식인 디디알(DDR,double data rate) 동기식 메모리 장치가 제안되었다.

<19> 디디알 동기식 메모리 장치의 각 데이터 입출핀에서는 외부에서 입력되는 클록의 상승 에지(rising edge)와 하강 에지(falling edge)에 동기되어 연속적으로 두 개의 데이터가 입출력되는 바, 클록의 주파수를 증가시키지 않더라도 종래의 SDR 동기식 메모리 장치에 비하여 최소한 두 배 이상의 대역폭(band width)을 구현할 수 있어 그 만큼 고속동작이 구현 가능하다.

<20> 그런데, 디디알 메모리 장치에서는 두 개의 데이터를 한 클럭 주기에서 내보내거나 또는 입력받아야 하기 때문에, 이를 효과적으로 수행하기 위해서는 종래의 동기식 메모리 장치에서 사용되고 있는 데이터 억세스 방식을 사용할 수가 없다.

<21> 만약 클럭의 주기(cycle)가 10nsec 정도라면 상승 및 하강시의 시간(약 $0.5 \times 4 = 2$)과 그 밖의 스펙을 맞추기 위한 시간 등을 빼면 실질적으로 약 6nsec 이하의 시간동안 두 개의 데이터를 연속적으로 처리하여야 하는데, 이러한 처리는 메모리 장치의 내부에서 수행하기에 역부족이므로, 메모리 장치는 외부로 데이터를 내보내거나 입력받을 때만 클럭의 라이징에지 및 폴링에지에서 데이터를 입출력시키고, 실질적으로 메모리 장치 내부에서는 클럭의 한쪽에지에 동기되는 두 개의 데이터로 처리된다.

<22> 따라서 메모리 장치에서 데이터를 입력받아 내부 코어영역으로 전달하거나, 코어영역에서 전달되는 데이터를 외부로 출력하기 위해서는 새로운 데이터 억세스 방식이 필요하다.

<23> 이를 위하여 디디알 메모리 장치의 데이터 입력버퍼는 상승에지 및 하강에지에 동기된 2비트(bit)의 데이터를 프리패치하고, 이를 메인클럭의 상승에지에 짹수데이터 또는 홀수데이터로 동기시켜 내부 코어영역으로 전달하고 있다.

<24> 한편, 데이터 입출력의 정확한 타이밍을 구현하기 위해 데이터를 입력받을 때 메모리장치 외부의 중앙처리장치(CPU)나 메모리 컨트롤러(controller)에서 데이터신호와 함께 데이터스트로브(data strobe) 신호(DQS)가 함께 입력된다.

<25> 도1은 종래기술에 의한 동기식 메모리 장치의 2비트 프리패치 데이터 입력버퍼를 나타내는 블럭구성도이다.

<26> 도1을 참조하여 살펴보면, 동기식 메모리 장치의 2비트 프리패치 데이터 입력버퍼는 라이트 명령에 의해 생성되는 인에이블신호(en_dinds)에 의해 인에이블되어 데이터스트로브 신호(DQS)의 라이징에지와 폴링에지에 각각 생성되는 라이징펄스(dsdp)와 폴링펄스(dsfp)를 출력하는 데이터스트로브 버퍼부와, 데이터를 외부로부터 입력받는 데이터 버퍼부와, 라이징펄스(dsdp)에 의해 데이터 버퍼부에서 출력되는 데이터(data)를 래치하는 라이징래치와, 폴링펄스(dsfp)에 의해 데이터 버퍼부에서 출력되는 데이터(data)를 래치하는 폴링래치와, 폴링펄스(dsfp)에 의해 라이징래치에서 출력되는 데이터신호(rising_data)를 래치하여 출력시킴으로서 폴링래치에서 출력되는 폴링데이터(falling_data)와 얼라인 출력데이터(align_dr)를 얼라인시키는 데이터얼라인부와, 얼라인 출력데이터(align_dr)와 폴링래치의 출력 데이터(falling_data)를 입력받아 내부 스트로브신호(data_strobe)에 응답하여 메모리 장치의 내부로 짹수 데이터 또는 홀수 데이터로 출력하는 글로벌 입출력라인 드라이버를 구비한다.

<27> 도2는 도1에 도시된 데이터 입력버퍼의 동작을 나타내는 타이밍도이다.

<28> 이하 도1 및 도2를 참조하여 2비트를 프리패치하는 데이터 입력버퍼의 동작을 살펴본다.

<29> 먼저 도2를 참조하여 살펴보면, 클럭(CLK)의 라이징에지와 폴링에지에 동기되어 데이터(D0 ~ D3)가 입력되고, 데이터가 입력되는 타이밍에 맞추어 데이터스트로버 신호(DQS)가 입력된다.

<30> 데이터스트로브 신호(DQS)는 평상시에는 하이임피던스 상태를 유지하고 있다가, 데이터가 입력되기 한 클럭전에 미리 로우 레벨을 유지하는 프리앰블(preamble)상태에서 데이터가 입력되는 타이밍에 따라서 클럭킹되다가, 데이터가 모두 입력되고 나면 다시 일정기간 로우레벨의 포스트앰블(postamble) 상태를 유지하다가 다시 하이 임피던스 상태를 유지하게 된다.

<31> 데이터스트로브 버퍼부는 라이트명령어에 의해 생성되는 인에이블신호(endinds)에 의해 인에이블되어 데이터스트로브 신호(DQS)의 라이징에지에서 펄스형태로 출력되는 라이징펄스(dsdp)와, 데이터스트로브 신호(DQS)의 폴링에지 펄스형태로 출력되는 폴링펄스(dsfp)를 생성하여 출력한다.

<32> 이어서 라이징래치는 제1 데이터와 제3 데이터(D0,D2)를 라이징펄스(dsdp)에 의해 래치하여 라이징데이터(rising_data)로 출력한다. 이어서 폴링래치는 제2 데이터와 제4 데이터(D1,D3)를 폴링펄스(dsfp)에 의해 래치하여 폴링데이터(falling_data)를 출력하고, 한편으로 데이터 얼라인부는 라이징데이터(rising_data)를 폴링펄스(dsfp)에 의해 다시 래치하여 얼라인데이터(align_data)로 출력한다. 여기서 데이터 얼라인부는 글로벌 입출력라인 드라이버로 전달되는 데이터의 얼라인을 위한 것이다.

<33> 이어서 글로벌 입출력라인 드라이버는 내부 스트로브신호(data_strobe)에 응답하여 얼라인데이터(align_data)와 폴링데이터(falling_data)를 글로벌 입출력라인(Global I/O)에 전달하게 된다. 이후 글로벌 입출력라인에 인가된 데이터는 입출력 센스앰프에 의해 감지 증폭된 다음 셀어레이로 전달된다.

<34> 그러나 중앙처리장치등의 반도체 장치가 더 고속화되면서 메모리 장치를 더 고속으로 동작시켜야 하는 요구가 생겼는데, 이를 위해 4비트의 데이터를 프리패치하여 메모리 장치의 내부로 전달하는 4비트 프리패치 데이터 입력버퍼가 제안되었다.

<35> 도3은 종래기술에 의한 동기식 메모리 장치의 4비트 프리패치 데이터 입력버퍼를 나타내는 블럭구성도이다.

<36> 도3을 참조하여 살펴보면, 동기식 메모리 장치의 4비트 프리패치 데이터 입력버퍼는 라이트 명령에 의해 생성되는 인에이블신호(en_dinds)에 의해 인에이블되어 데이터스트로브 신호(DQS)의 라이징에지와 폴링에지에 각각 생성되는 라이징펄스(dsdp4)와 폴링펄스(dsfp4)를 출력하는 데이터스트로브 버퍼부(190)와, 데이터를 외부로부터 입력받는 데이터버퍼부(100)와, 라이징펄스(dsdp4)에 의해 데이터 버퍼부(100)에서 출력되는 데이터(data)를 래치하여 제1 라이징데이터(rising_d0)로 출력하는 제1 라이징래치(110)와, 제1 라이징데이터(rising_d0)를 폴링펄스(dsfp4)에 의해 래치하여 제3 얼라인데이터(align_dr1)를 출력하는 제2 라이징래치(120)와, 라이징펄스(dsdp4)에 의해 제3 얼라인데이터(align_dr1)를 래치하여 제2 라이징펄스(rising_d1)를 출력하는 제3 라이징래치(140)와, 제2 라이징데이터(rising_d1)를 폴링펄스(dsfp4)에 의해 래치하여 제1 얼라인데이터(align_r0)를 출력하는 제4 라이징래치(140)와, 폴링펄스(dsfp4)에 의해 데이터버퍼부(100)에서 출력되는 데이터(data)를 래치하여 제4 얼라인데이터(align_df1)로 출력하는 제1 폴링래치(130)와, 제4 얼라인데이터(align_df1)를 라이징펄스(dsdp4)에 의해 래치하여 폴링데이터(falling_d1)를 출력하는 제2 폴링래치(150)와, 폴링데이터(falling_d1)를 폴링펄스(dsfp4)에 의해 래치하여 제2 얼라인데이터(align_df0)로 출력하는 제3 폴링래치(150)와, 제1 내지 제4 얼라인데이터(align_dr0, align_df0, align_dr1, align_df1)를 입력받아 내부 스트로브신호(data_strobe)에 응

답하여 글로벌 입출력라인(GIO)으로 짹수데이터(gid_ev0, gio_ev1) 또는 홀수 데이터(gid_od0, gio_od1)로 출력하는 글로벌 입출력라인 드라이버(180)를 구비한다.

<37> 도4는 도3에 도시된 데이터 입력버퍼의 동작을 나타내는 타이밍도이다.

<38> 이하 도3 및 도4를 참조하여 4비트를 프리패치하는 데이터 입력버퍼의 동작을 살펴본다.

<39> 먼저 도4를 참조하여 살펴보면, 클럭(CLK)의 라이징에지와 폴링에지에 동기되어 데이터(D0 ~ D7)가 입력되고, 데이터가 입력되는 타이밍에 맞추어 데이터스트로브 신호(DQS)가 입력된다.

<40> 데이터스트로브 버퍼부(190)는 라이트명령어에 의해 생성되는 인에이블신호(endinds)에 의해 인에이블되어 데이터스트로브 신호(DQS)의 라이징에지에서 펠스형태로 출력되는 라이징펠스(dsdp4)와, 데이터스트로브 신호(DQS)의 폴링에지 펠스형태로 출력되는 폴링펠스(dsfp4)를 생성하여 출력한다.

<41> 이어서 제1 라이징래치(110)는 제1,3,5,7 데이터(D0,D2,D4,D6)를 라이징펠스(dsdp4)에 의해 래치하여 제1 라이징데이터(rising_d0)로 출력한다.

<42> 이어서 제2 라이징래치(120)는 폴링펠스(dsfp4)에 의해 제1 라이징데이터(rising_d0)를 래치하여 제3 얼라인데이터(align_r1)를 출력하고, 제1 폴링래치(130)는 폴링펠스(dsfp4)에 의해 제2,4,6,8 데이터(D1,D3,D5,D7)를 래치하여 제4 얼라인데이터(align_f1)로 출력한다.

<43> 이어서, 제3 라이징래치(140)는 라이징펠스(dsdp4)에 의해 제3 얼라인데이터(align_r1)를 래치하여 제2 라이징데이터(160)로 출력하고, 제2 폴링래치(150)는 라이징펠스(dsfp4)에 의해 제4 얼라인데이터(align_f1)을 래치하여 폴링데이터(falling_d1)로 출력한다.

<44> 이어서 제4 라이징래치(160)는 폴링펄스(dsfp4)에 의해 제2 라이징데이터(rising_d1)를 래치하여 제1 얼라인데이터(align_r0)로 출력하고, 제3 폴링래치(170)은 폴링펄스(fsfp4)에 의해 폴링데이터(align_df0)를 래치하여 제2 얼라인데이터(align_f0)로 출력한다.

<45> 이어서 글로벌 입출력라인 드라이버(180)는 내부 스트로브신호(data_strobe)에 응답하여 제1 내지 제4 얼라인데이터(align_dr0, align_df0, align_dr1, align_df1)를 짹수데이터(gid_ev0, gio_ev1) 또는 홀수 데이터(gid_od0, gio_od1)로 글로벌 입출력라인(Global I/O)에 전달하게 된다.

<46> 도4에 도시된 바와 같이 내부 스트로브신호(data_strobe)가 글로벌 입출력라인 드라이버(180)에 입력되면, 그 때 얼라인되어 입력되는 제1 내지 제4 얼라인데이터(align_dr0, align_df0, align_dr1, align_df1)를 글로벌 입출력라인(gio 라인)으로 전달하게 되는 것이다. 따라서 글로벌 입출력라인으로 얼라인된 4비트의 데이터를 바르게 내부코어영역으로 전달하기 위해서는 내부 스트로브신호(data_strobe)는 구간 'X'내에서 글로벌 입출력라인 드라이버로 출력되어야 한다.

<47> 도5는 도4에 도시된 4비트를 프리패치하는 데이터 입력버퍼의 문제점을 나타내는 타이밍도이다. 이어서 도5를 참조하여 종래기술에 의한 4비트를 프리패치하는 데이터 입력버퍼의 문제점을 살펴본다.

<48> 메모리 장치에서 라이트명령이 입력된 후에 데이터가 입력되는 타이밍에 따라서 입력되는 데이터스트로브 신호(DQS)는 $(WL-0.25) \times CK \sim (WL+0.25) \times CK$ 의 마진을 가지고 입력이 된다. 여기서 WL은 라이트 레이턴시(Write latency)를 뜻하는 것으로 라이트명령어가 입력된 후 데이터가 입력될 때까지의 타이밍을 나타내는 것이다.

<49> 따라서 데이터가 입력되는 타이밍에 입력되는 데이터스트로브 신호(DQS)는 약 $0.5t_{CK}$ 의 마진을 가지고 입력되는 것이다. 즉, 만약 $WL=1$ 이라면, 메모리 장치가 동작하는 어떤 순간에는 라이트명령어가 입력된 뒤에 $0.75 \times t_{CK}$ 이후에 데이터스트로브 신호(DQS)가 입력되고, 또한 다른 어떤 순간에는 라이트명령어가 입력된 뒤에 $1.75 \times t_{CK}$ 이후에 데이터스트로브 신호(DQS)가 입력될 수 있는 것이다.

<50> 도5에 도시된 DQS1은 라이트명령어가 입력된 뒤에 $0.75 \times t_{CK}$ 이후에 입력되는 데이터스트로브 신호를 나타내는 것이고, DQS2는 라이트명령어가 입력된 뒤에 $1.75 \times t_{CK}$ 이후에 입력되는 데이터스트로브 신호(DQS)를 나타낸다.

<51> 따라서 DQS1에 의해 래치된 데이터(D_0, D_2, \dots)와 DQS2에 래치된 데이터(D_0, D_2, \dots)는 $0.5 \times t_{CK}$ 의 간격을 가지고 있다.

<52> 얼라인된 데이터가 글로벌 입출력라인으로 정확하게 전달될 수 있게 하기 위해서는 내부 스트로브신호(data_strobe)는 DQS1과 DSQ2에 의해 래치된 데이터가 함께 공유하는 구간에 반드시 생성되어 글로벌 입출력라인 드라이버로 입력이 되어야한다.

<53> 따라서 동기식 메모리 장치에서 데이터를 입력받아 얼라인시킬 수 있는 마진이 $0.5 \times t_{CK}$ 인 것이다. 만약 500MHz로 동작하는 메모리 장치에서 동작클럭의 주기는 $t_{CK}=2n$ 가 되는데, 이 때의 내부 데이터스트로브 신호(data_strobe)가 입력될 수 있는 마진은 $1.0n$ 이 되는 것이다.

<54> 이 경우 내부 스트로브 신호(data_strobe)의 전 후 마진을 생각했을 때, 내부 스트로브 신호(data_strobe)의 마진은 $0.5n$ 정도밖에 되지 않아서 얼라인된 데이터를 내부 글로벌 입출력라인에 제대로 전달할 수 없는 경우가 발생하게 된다.

<55> 메모리 장치는 점점 더 고속화됨에 따라서 내부 스트로브신호(data_strobe)의 마진은 점점 더 줄어들수 밖에 없고, 이로 인해 라이트명령어에 대응하여 입력된 데이터를 열라인하여 내부 코어로 신뢰성있게 전달하기가 점점 더 어려워지는 것이다.

【발명이 이루고자 하는 기술적 과제】

<56> 본 발명은 상기의 문제점을 해결하기 위해 제안된 것으로, 클럭에 동기되어 입력되는 데 이터를 얼라인하여 내부회로로 전달하는데 있어서의 데이터 얼라인 마진을 증가시켜 고주파에서도 안정적으로 데이터를 입력받아 처리할 수 있는 메모리 장치를 제공함을 목적으로 한다.

【발명의 구성 및 작용】

인시킨 제3 및 제4 얼라인데이터를 출력하는 제3 래치수단; 및 상기 제1 내지 제4 얼라인데이터를 짹수데이터 또는 홀수데이터로 선택하여 출력하기 위한 글로벌 입출력라인 드라이버를 구비하는 동기식 메모리 장치를 제공한다.

<58> 또한 본 발명은 동작클럭의 라이징에지와 폴링에지에 동기되어 다수의 데이터를 입력받는 동기식 메모리 장치에 있어서, 상기 데이터가 입력되는 타이밍에 클럭킹되는 데이터스트로브 신호를 입력받아. 홀수번째 입력된 데이터스트로브 신호의 라이징에지와 폴링에지를 각각 검출하는 제1 라이징펄스 및 제1 폴링펄스와, 짹수번째 입력된 데이터스트로브 신호의 라이징에지와 폴링에지를 각각 검출하는 제2 라이징펄스와 제2 폴링펄스를 출력하기 위한 데이터스트로브 버퍼링 수단; 상기 제1 라이징펄스에 동기되도록, 상기 제1 데이터를 래치시키는 제1 라이징래치; 상기 제1 폴링펄스에 동기되도록, 상기 제1 라이징래치에 의해 래치된 제1 데이터를 다시 래치시키는 제2 라이징래치; 상기 제1 폴링펄스에 동기되도록, 상기 제2 데이터를 래치시키는 제1 폴링래치; 상기 제2 폴링펄스에 동기되도록, 상기 제2 라이징래치에 의해 래치된 제1 데이터를 다시 래치시켜 상기 제1 얼라인데이터로 출력하는 제3 라이징래치; 상기 제2 폴링펄스에 동기되도록, 상기 제1 폴링래치에 의해 래치된 데이터를 다시 래치시켜 상기 제2 얼라인데이터로 출력하는 제2 폴링래치; 상기 제2 라이징펄스에 동기되도록, 상기 제3 데이터를 래치시키는 제4 라이징래치; 상기 제2 폴링펄스에 동기되도록, 상기 제4 라이징래치에 래치된 데이터를 다시 래치시켜 상기 제3 얼라인데이터로 출력하는 제5 라이징래치; 및 상기 제2 폴링펄스에 동기되도록, 상기 제4 데이터를 래치시켜 상기 제4 얼라인데이터로 출력하는 제3 폴링래치; 및 상기 제1 내지 제4 얼라인데이터를 짹수데이터 또는 홀수데이터로 선택하여 출력하기 위한 글로벌 입출력라인 드라이버를 구비하는 메모리 장치를 제공한다.

<59> 이하, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자가 본 발명의 기술적 사상을 용이하게 실시할 수 있을 정도로 상세히 설명하기 위하여, 본 발명의 가장 바람직한 실시예를 첨부된 도면을 참조하여 설명하기로 한다.

<60> 도6은 본 발명의 바람직한 실시예에 따른 동기식 메모리 장치의 4비트 프리패치 데이터 입력버퍼를 나타내는 블럭구성도이다.

<61> 도6을 참조하여 살펴보면, 본 실시예에 따른 동기식 메모리 장치의 4비트 프리패치 데이터 입력버퍼는 동작클럭의 라이징에지와 폴링에지에 동기되어 다수의 데이터가 입력되는 타이밍에 클럭킹되는 데이터스트로브 신호(DQS)를 입력받아. 홀수번째 입력된 데이터스트로브 신호(DQS)의 라이징에지와 폴링에지를 각각 검출하는 제1 라이징펄스(dsdp4_1) 및 제1 폴링펄스(dsfp4_1)와, 짹수번째 입력된 데이터스트로브 신호(DQS)의 라이징에지와 폴링에지를 각각 검출하는 제2 라이징펄스(dsdp4_2)와 제2 폴링펄스(dsfp4_2)를 출력하기 위한 데이터스트로브 버퍼부(300)와, 제1 폴링펄스(dsfp4_1)에 동기되도록 동작클럭의 라이징에지에 입력되는 제1 데이터와 동작클럭의 폴링에지에 입력되는 제2 데이터를 얼라인시키는 제1 래치(210)와, 제2 폴링펄스(dsfp4_2)에 동기되도록 제1 래치(210)에 얼라인된 제1 및 제2 데이터(rising_d1, falling_d1)를 재얼라인시킨 제1 및 제2 얼라인데이터(align_dr0, align_df0)를 출력하는 제2 래치(220)와, 제2 폴링펄스(dsfp4_2)에 동기되도록 상기 제2 데이터에 이어서 입력되는 제3 데이터와 제4 데이터를 얼라인시킨 제3 및 제4 얼라인데이터(align_dr1, align_f1)를 출력하는 제3 래치(230)와, 제1 내지 제4 얼라인데이터(align_dr0, align_df0, align_dr1, align_df1)를 짹수데이터(gio_ev0, gio_ev1) 또는 홀수데이터(gio_od0, gio_od1)로 선택하여 출력하기 위한 글로벌 입출력라인 드라이버(290)을 구비한다.

<62> 또한, 제1 래치(210)는 제1 라이징펄스(dsdp4_1)에 동기되도록 상기 제1 데이터를 래치시키는 제1 라이징래치(211)와, 제1 폴링펄스(dsfp4_1)에 동기되도록 제1 라이징래치(211)에 의해 래치된 제1 데이터를 다시 래치시키는 제2 라이징래치(212)와, 제1 폴링펄스(dsfp4_1)에 동기되도록 상기 제2 데이터를 래치시키는 제1 폴링래치(213)를 구비한다.

<63> 또한, 제2 래치(220)는 제2 폴링펄스(dsfp4_2)에 동기되도록 제2 라이징래치(212)에 의해 래치된 제1 데이터를 다시 래치시켜 제1 얼라인데이터(align_dr0)로 출력하는 제3 라이징래치(221)와, 제2 폴링펄스(dsfp4_2)에 동기되도록 제1 폴링래치(213)에 의해 래치된 데이터를 다시 래치시켜 제2 얼라인데이터(align_df0)로 출력하는 제2 폴링래치(222)를 구비한다.

<64> 또한, 제3 래치(230)은 제2 라이징펄스(dsdp4_2)에 동기되도록 상기 제3 데이터를 래치시키는 제4 라이징래치(231)와, 제2 폴링펄스(dsfp4_2)에 동기되도록 제4 라이징래치(231)에 래치된 데이터를 다시 래치시켜 제3 얼라인데이터(align_dr1)로 출력하는 제5 라이징래치(232)와, 제2 폴링펄스(dsfp4_2)에 동기되도록 상기 제4 데이터를 래치시켜 제4 얼라인데이터(align_df1)로 출력하는 제3 폴링래치(233)를 구비한다.

<65> 도7은 도6에 도시된 데이터스트로브 버퍼부를 나타내는 블럭구성도이다.

<66> 도7을 참조하여 살펴보면, 데이터스트로브 버퍼부(300)는 라이트 명령에 의해 생성되는 라이트펄스(wtp)를 출력하는 명령어디코더(310)와, 라이트펄스(wtp)에 의해 초기 동작모드가 셋팅되며, 데이터스트로브 신호(DQS)를 이용하여 제1 및 제2 라이징펄스(dsdp4_1, dsdp4_2)와 제1 및 제2 폴링펄스(dsfp4_1, dsfp4_2)를 생성하여 출력하기 위한 데이터스트로브 디바이더(340)를 구비한다.

<67> 또한 데이터 스트로브 버퍼부(340)는 라이트펄스(wtp)를 라이트레이턴시(WL)보다 한 클럭 적은 구간((WL-1)×CK)만큼 시프팅(shifting)시켜 데이터스트로브 디바이더(340)로 출력하는 레이턴시 시프터(320)를 더 구비한다.

<68> 또한 데이터 스트로브 버퍼부(340)는 데이터스트로브 신호(DQS)를 입력 받아 버퍼링하여 데이터스트로브 디바이더(340)로 출력하는 DQS버퍼부(330)를 더 구비한다.

<69> 도8은 도7에 도시된 데이터스트로브 디바이더의 일실시예를 나타내는 회로도이다.

<70> 도8을 참조하여 살펴보면, 데이터스트로브 디바이더(340)는 초기동작 셋팅을 위한 초기 동작 셋팅부(341)과, 홀수번째 데이터스트로브 신호(DQS)의 라이징을 감지하여 제1 라이징펄스(dsdp4_1)를 출력하는 제1 펄스생성부(342)와, 홀수번째 데이터스트로브 신호(DQS)의 폴링을 감지하여 제1 폴링펄스(dsfp4_1)를 출력하는 제2 펄스생성부(343)와, 짹수번째 데이터스트로브 신호(DQS)의 라이징을 감지하여 제2 라이징펄스(dsdp4_2)를 출력하는 제3 펄스생성부(343)와, 짹수번째 데이터스트로브 신호(DQS)의 폴링을 감지하여 제2 폴링펄스(dsfp4_2)를 출력하는 제4 펄스생성부(344)를 구비한다.

<71> 도9는 도6에 도시된 데이터 입력버퍼의 동작을 나타내는 타이밍도이다. 이하 도6 내지 도9를 참조하여 본 실시예에 따른 메모리 장치의 데이터 입력버퍼의 동작을 살펴본다.

<72> 먼저 도9를 참조하여 살펴보면, 클럭(CLK)의 라이징에지와 폴링에지에 동기되어 데이터 (D0 ~ D7)가 입력되어 데이터버퍼부(200)에 의해 버퍼링되고, 데이터가 입력되는 타이밍에 맞추어 데이터스트로브 신호(DQS)가 클럭킹되어 입력된다.

<73> 데이터스트로브 버퍼부(300)는 라이트명령어에 의해 생성되는 인에이블신호(endinds)에 의해 인에이블되고, 데이터스트로브 신호(DQS)를 입력받아서, 홀수번째 데이터스트로브 신호

(DQS)의 라이징에지와 폴링에지에서 각각 펄스형태로 출력되는 제1 라이징펄스(dsdp4_1) 및 제2 폴링펄스(dsfp4_1)와, 짹수번째 데이터스트로브 신호(DQS)의 라이징에지와 폴링에지에서 각각 펄스형태로 출력되는 제2 라이징펄스 및 제2 폴링펄스를 생성하여 출력한다.

<74> 이어서 제1 라이징래치(211)는 데이터버퍼부(200)에서 출력되는 데이터(data) 중에서 제1 데이터(D0)와 제5 데이터(D4)를 제1 라이징펄스(dsdp4_1)에 의해 래치하여 제1 라이징데이터(rising_d0)로 출력한다.

<75> 이어서 제2 라이징래치(212)는 제1 폴링펄스(dsfp4_1)에 의해 제1 라이징데이터(rising_d0)를 다시 래치하여 제2 라이징데이터(rising_d1)로 출력하고, 제1 폴링래치(213)는 데이터버퍼부(200)에서 출력되는 데이터(data) 중에서 제1 폴링펄스(dsfp4_1)에 의해 제2 데이터(D1)와 제6 데이터(D5)를 래치하여 폴링데이터(falling_d1)로 출력한다.

<76> 이어서 제3 라이징래치(221)는 제2 폴링펄스(dsfp4_2)에 의해 제2 라이징데이터(rising_d1)를 래치하여 제1 얼라인데이터(align_dr0)로 출력하고, 제2 폴링래치(222)는 제2 폴링펄스(dsfp4_2)에 의해 폴링데이터(falling_d1)를 래치하여 제2 얼라인데이터(align_df0)로 출력한다.

<77> 한편, 제4 라이징래치(231)는 제2 라이징펄스(dsdp4_2)에 의해 데이터버퍼부(200)에서 출력되는 데이터(data)중에서 제3 데이터(D2)와 제7 데이터(D6)를 래치하여 제2 라이징데이터(rising_d2)로 출력한다.

<78> 이어서 제5 라이징래치(232)는 제2 폴링펄스(dsfp4_2)에 의해 제2 라이징데이터(rising_d2)를 다시 래치하여 제3 얼라인데이터(align_dr1)로 출력하고, 제3 폴링래치(233)는

데이터버퍼부(200)에서 출력되는 데이터(data)중에서 제4 데이터(D3)와 제8 데이터(D7)를 래치하여 제4 얼라인데이터(align_df1)로 출력한다.

<79> 이어서 글로벌 입출력라인 드라이버(290)는 내부 스트로브신호(data_strobe)에 응답하여 제1 내지 제4 얼라인데이터(align_dr0, align_df0, align_dr1, align_df1)를 짹수데이터(gid_ev0, gio_ev1) 또는 홀수 데이터(gid_od0, gio_od1)로 글로벌 입출력라인(Global I/O)에 전달하게 된다.

<80> 도9에 도시된 바와 같이 내부 스트로브신호(data_strobe)가 글로벌 입출력라인 드라이버(290)에 입력되면, 그 때 얼라인되어 입력되는 제1 내지 제4 얼라인데이터(align_dr0, align_df0, align_dr1, align_df1)를 글로벌 입출력라인으로 전달하게 되는 것이다. 따라서 글로벌 입출력라인으로 얼라인된 4비트의 데이터를 바르게 전달하기 위해서는 내부 스트로브신호(data_strobe)는 구간 'Y'내에서 글로벌 입출력라인 드라이버로 출력되어야 한다.

<81> 도10은 도6에 도시된 데이터 입력버퍼의 데이터 얼라인 마진을 나타내는 타이밍도이다. 이하 도10을 참조하여 본 실시예에 따른 데이터 입력버퍼의 데이터 얼라인 마진을 살펴본다.

<82> 전술한 바와 같이, 동작클럭의 라이징에지와 폴링에지에 동기되어 데이터가 입력될 때에, 같이 입력되는 데이터스트로브 신호(DQS)는 $(WL-0.25) \times CK \sim (WL+0.25) \times CK$ 의 마진을 가지고 입력이 된다. 여기서 WL은 라이트 레이턴시(Write latency)를 뜻하는 것으로 라이트명령어가 입력된 후 데이터가 입력될 때까지의 타이밍을 나타내는 것이다.

<83> 따라서 데이터가 입력되는 타이밍에 입력되는 데이터스트로브 신호(DQS)는 약 $0.5tCK$ 의 마진을 가지고 입력되는 것이다. 즉, 만약 $WL=1$ 이라면, 메모리 장치가 동작하는 어떤 순간에는 라이트명령어가 입력된 뒤에 $0.75 \times CK$ 이후에 데이터

스트로브 신호(DQS)가 입력되고, 또한 다른 어떤 순간에는 라이트명령어가 입력된 뒤에 $1.75 \times tCK$ 이후에 데이터스트로브 신호(DQS)가 입력될 수 있는 것이다.

<84> 도10에 도시된 DQS1은 라이트명령어가 입력된 뒤에 $0.75 \times tCK$ 이후에 입력되는 데이터스트로브 신호를 나타내는 것이고, DQS2는 라이트명령어가 입력된 뒤에 $1.75 \times tCK$ 이후에 입력되는 데이터스트로브 신호(DQS)를 나타낸다. 따라서 DQS1에 의해 래치된 데이터(D0,D2,...)와 DQS2에 래치된 데이터(D0,D2,...)는 $0.5 \times tCK$ 의 간격을 가지고 있다.

<85> 얼라인된 데이터가 글로벌 입출력라인으로 정확하게 전달될 수 있게 하기 위해서는 내부 스트로브신호(data_strobe)는 DQS1과 DSQ2에 의해 래치된 데이터가 함께 공유하는 구간에 반드시 생성되어 글로벌 입출력라인 드라이버로 입력이 되어야한다.

<86> 도시된 바와 같이 DQS1과 DSQ2에 의해 래치된 데이터가 함께 공유하는 구간은 $0.5 \times tCK$ 이 된다. 따라서 내부 스트로브신호(data_strobe)는 $0.5 \times tCK$ 이 구간안에 글로벌입출력라인 드라이버로 출력되면 되는 것이다.

<87> 만약 500MHz로 동작하는 메모리 장치의 한클럭의 주기는 $tCK=2n$ 인데, 이 때의 내부 데이터스트로브 신호(data_strobe)의 입력 마진은 3.0n가 되는 것이다.

<88> 이는 종래기술에서 동기식 메모리 장치에 데이터가 입력되어 얼라인 시킬 수 있는 마진이 $0.5 \times tCK$ 이던 것에 비해서, 본 발명에 의해서 동기식 메모리 장치에 데이터가 입력되어 얼라인 시킬 수 있는 마진이 3배로 증가되는 것을 나타내는 것이다.

<89> 데이터가 입력되어 얼라인 시킬 수 있는 마진이 증가함으로서 메모리 장치가 데이터를 안정적으로 내부코어로 전달할 수 있게 된다. 또한, 차후 더 고속으로 동작하는 동기식 메모리

장치를 개발할 때도 입력된 데이터를 열라인시키는 마진이 충분하여 현재의 데이터 입력버퍼를 그대로 사용할 수 있다.

<90> 이상에서 설명한 본 발명은 전술한 실시예 및 첨부된 도면에 의해 한정되는 것이 아니고, 본 발명의 기술적 사상을 벗어나지 않는 범위 내에서 여러 가지 치환, 변형 및 변경이 가능하다는 것이 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 있어 명백할 것이다.

【발명의 효과】

<91> 본 발명에 의해서 입력되는 데이터의 열라인 마진을 종래의 메모리 장치보다 3배이상으로 확보할 수 있기 때문에, 고속으로 동작하는 메모리 장치에서도 안정적으로 데이터를 입력받아 억세스할 수 있다.

【특허 청구범위】**【청구항 1】**

동작클럭의 라이징에지와 폴링에지에 동기되어 다수의 데이터를 입력받는 동기식 메모리 장치에 있어서,

상기 데이터가 입력되는 타이밍에 클럭킹되는 데이터스트로브 신호를 입력받아, 홀수번째 입력된 데이터스트로브 신호의 라이징에지와 폴링에지를 각각 검출하는 제1 라이징펄스 및 제1 폴링펄스와, 짹수번째 입력된 데이터스트로브 신호의 라이징에지와 폴링에지를 각각 검출하는 제2 라이징펄스와 제2 폴링펄스를 출력하기 위한 데이터스트로브 버퍼링 수단;

상기 제1 폴링펄스에 동기되도록, 상기 동작클럭의 라이징에지에 입력되는 제1 데이터와 상기 동작클럭의 폴링에지에 입력되는 제2 데이터를 얼라인시키는 제1 래치수단;

상기 제2 폴링펄스에 동기되도록, 상기 제1 래치수단에 얼라인된 제1 및 제2 데이터를 재얼라인시킨 제1 및 제2 얼라인데이터를 출력하는 제2 래치수단;

상기 제2 폴링펄스에 동기되도록, 상기 제2 데이터에 이어서 입력되는 제3 데이터와 제4 데이터를 얼라인시킨 제3 및 제4 얼라인데이터를 출력하는 제3 래치수단; 및

상기 제1 내지 제4 얼라인데이터를 짹수데이터 또는 홀수데이터로 선택하여 출력하기 위한 글로벌 입출력라인 드라이버
를 구비하는 동기식 메모리 장치.

【청구항 2】

제 1 항에 있어서,

상기 제1 래치수단은

상기 제1 라이징펄스에 동기되도록, 상기 제1 데이터를 래치시키는 제1 라이징 래치;

상기 제1 폴링펄스에 동기되도록, 상기 제1 라이징래치에 의해 래치된 제1 데이터를 다시 래치시키는 제2 라이징래치; 및

상기 제1 폴링펄스에 동기되도록, 상기 제2 데이터를 래치시키는 제1 폴링래치를 구비하는 것을 특징으로 하는 동기식 메모리 장치.

【청구항 3】

제 2 항에 있어서,

상기 제2 래치수단은

상기 제2 폴링펄스에 동기되도록, 상기 제2 라이징래치에 의해 래치된 제1 데이터를 다시 래치시켜 상기 제1 열라인데이터로 출력하는 제3 라이징래치; 및

상기 제2 폴링펄스에 동기되도록, 상기 제1 폴링래치에 의해 래치된 데이터를 다시 래치시켜 상기 제2 열라인데이터로 출력하는 제2 폴링래치를 구비하는 것을 특징으로 하는 동기식 메모리 장치.

【청구항 4】

제 1 항에 있어서,

상기 제3 래치수단은

상기 제2 라이징펄스에 동기되도록, 상기 제3 데이터를 래치시키는 제1 라이징 래치;

상기 제2 폴링펄스에 동기되도록, 상기 제1 라이징래치에 래치된 데이터를 다시 래치시켜 상기 제3 얼라인데이터로 출력하는 제2 라이징래치; 및
상기 제2 폴링펄스에 동기되도록, 상기 제4 데이터를 래치시켜 상기 제4 얼라인데이터로 출력하는 폴링래치를 구비하는 것을 특징으로 하는 동기식 메모리 장치.

【청구항 5】

제 1 항에 있어서,
상기 데이터스트로브 버퍼링 수단은
라이트 명령에 의해 생성되는 라이트펄스를 출력하는 명령어디코더; 및
상기 라이트펄스에 의해 초기동작모드가 셋팅되며, 상기 데이터스트로브 신호를 이용하여 상기 제1 및 제2 라이징펄스와 상기 제1 및 제2 폴링펄스를 생성하여 출력하기 위한 데이터스트로브 디바이더를 구비하는 것을 특징으로 하는 동기식 메모리 장치.

【청구항 6】

제 5 항에 있어서,
상기 데이터스트로브 버퍼링 수단은
상기 라이트펄스를 라이트레이턴시보다 한클럭 적은 구간만큼 시프팅시켜 상기 데이터스트로브 디바이더로 출력하는 레이턴시 시프터를 더 구비하는 것을 특징으로 하는 동기식 메모리 장치.

【청구항 7】

제 6 항에 있어서,

상기 데이터스트로브 버퍼링 수단은

상기 데이터스트로브 신호를 입력받아 버퍼링하여 상기 데이터스트로브 디바이더로 출력하는 데이터스트로브 버퍼를 더 구비하는 것을 특징으로 하는 동기식 메모리 장치.

【청구항 8】

동작클럭의 라이징에지와 폴링에지에 동기되어 다수의 데이터를 입력받는 동기식 메모리 장치에 있어서,

상기 데이터가 입력되는 타이밍에 클럭킹되는 데이터스트로브 신호를 입력받아. 홀수번째 입력된 데이터스트로브 신호의 라이징에지와 폴링에지를 각각 검출하는 제1 라이징펄스 및 제1 폴링펄스와, 짹수번째 입력된 데이터스트로브 신호의 라이징에지와 폴링에지를 각각 검출하는 제2 라이징펄스와 제2 폴링펄스를 출력하기 위한 데이터스트로브 버퍼링 수단;

상기 제1 라이징펄스에 동기되도록, 상기 제1 데이터를 래치시키는 제1 라이징래치;

상기 제1 폴링펄스에 동기되도록, 상기 제1 라이징래치에 의해 래치된 제1 데이터를 다시 래치시키는 제2 라이징래치;

상기 제1 폴링펄스에 동기되도록, 상기 제2 데이터를 래치시키는 제1 폴링래치;

상기 제2 폴링펄스에 동기되도록, 상기 제2 라이징래치에 의해 래치된 제1 데이터를 다시 래치시켜 상기 제1 얼라인데이터로 출력하는 제3 라이징래치;

상기 제2 폴링펄스에 동기되도록, 상기 제1 폴링래치에 의해 래치된 데이터를 다시 래치시켜 상기 제2 얼라인데이터로 출력하는 제2 폴링래치;

상기 제2 라이징펄스에 동기되도록, 상기 제3 데이터를 래치시키는 제4 라이징래치;

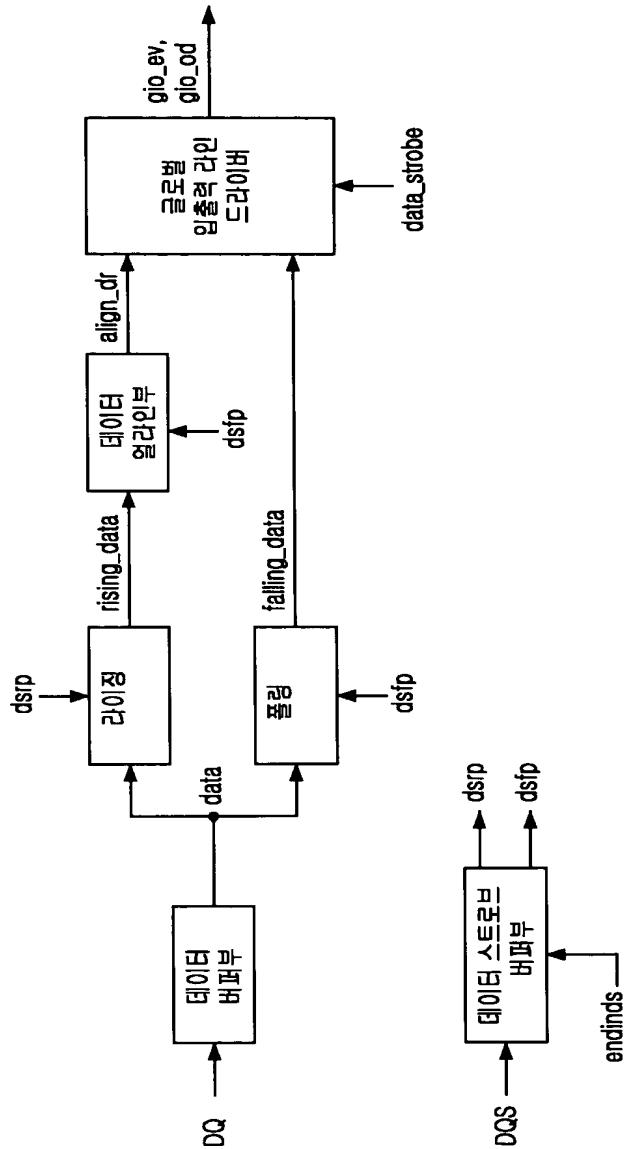
상기 제2 폴링펄스에 동기되도록, 상기 제4 라이징래치에 래치된 데이터를 다시 래치시켜 상기 제3 얼라인데이터로 출력하는 제5 라이징래치;

상기 제2 폴링펄스에 동기되도록, 상기 제4 데이터를 래치시켜 상기 제4 얼라인데이터로 출력하는 제3 폴링래치; 및

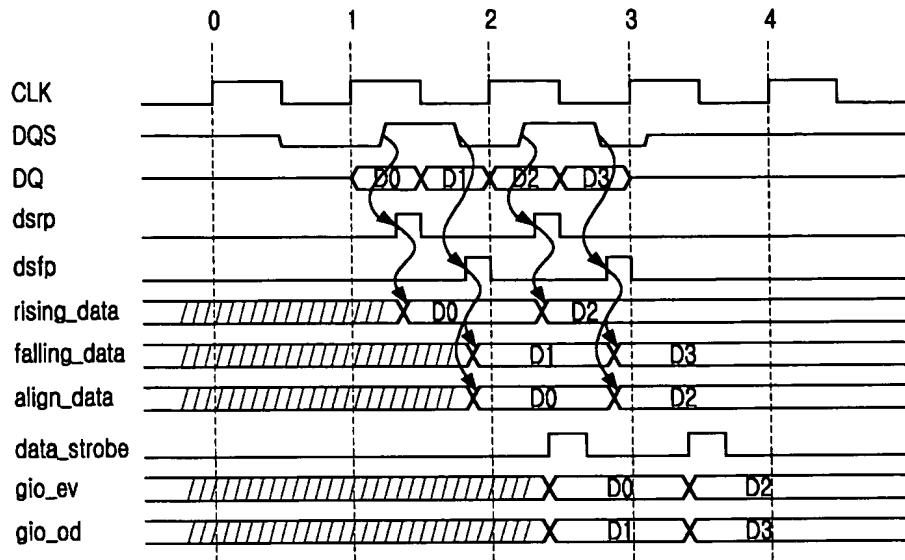
상기 제1 내지 제4 얼라인데이터를 짹수데이터 또는 홀수데이터로 선택하여 출력하기 위한 글로벌 입출력라인 드라이버를 구비하는 동기식 메모리 장치.

【도면】

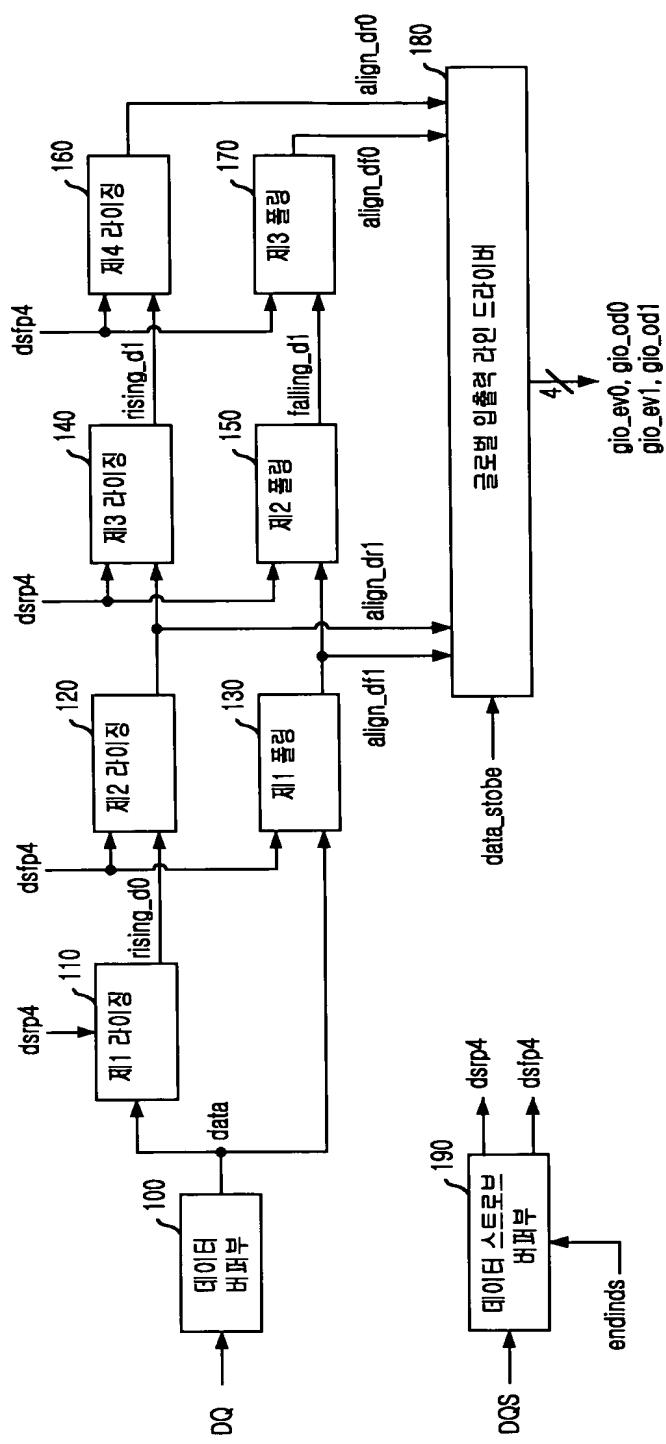
【도 1】



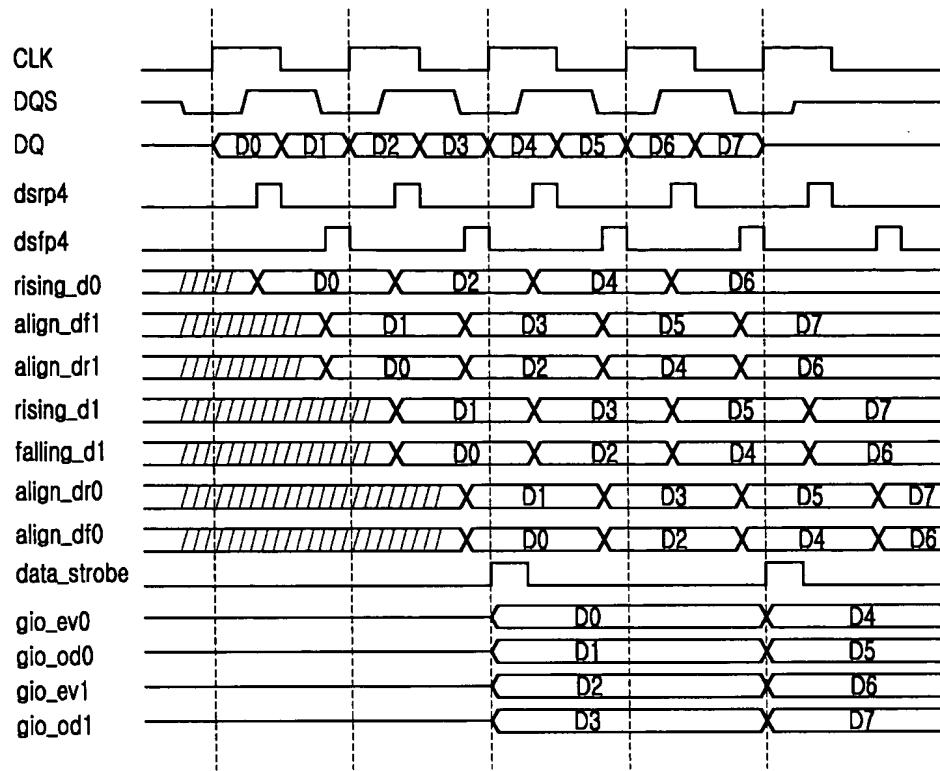
【도 2】



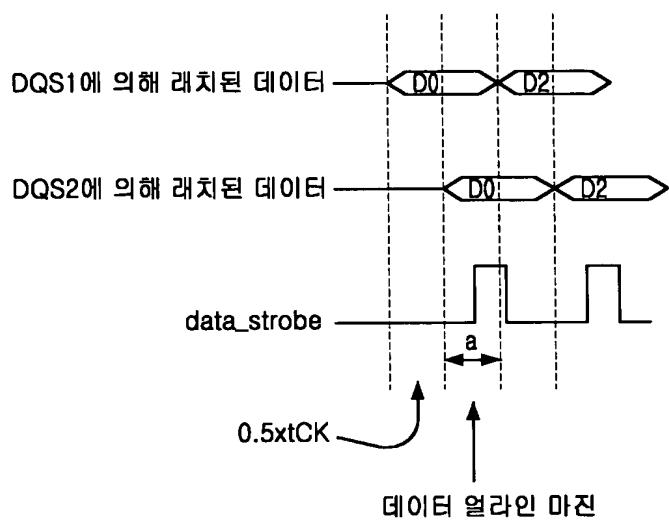
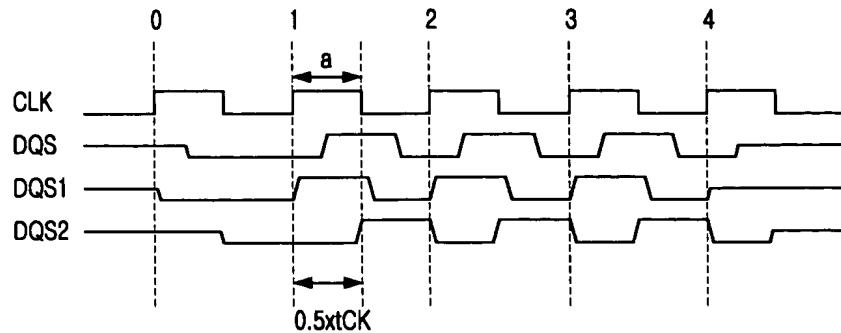
【H 3】



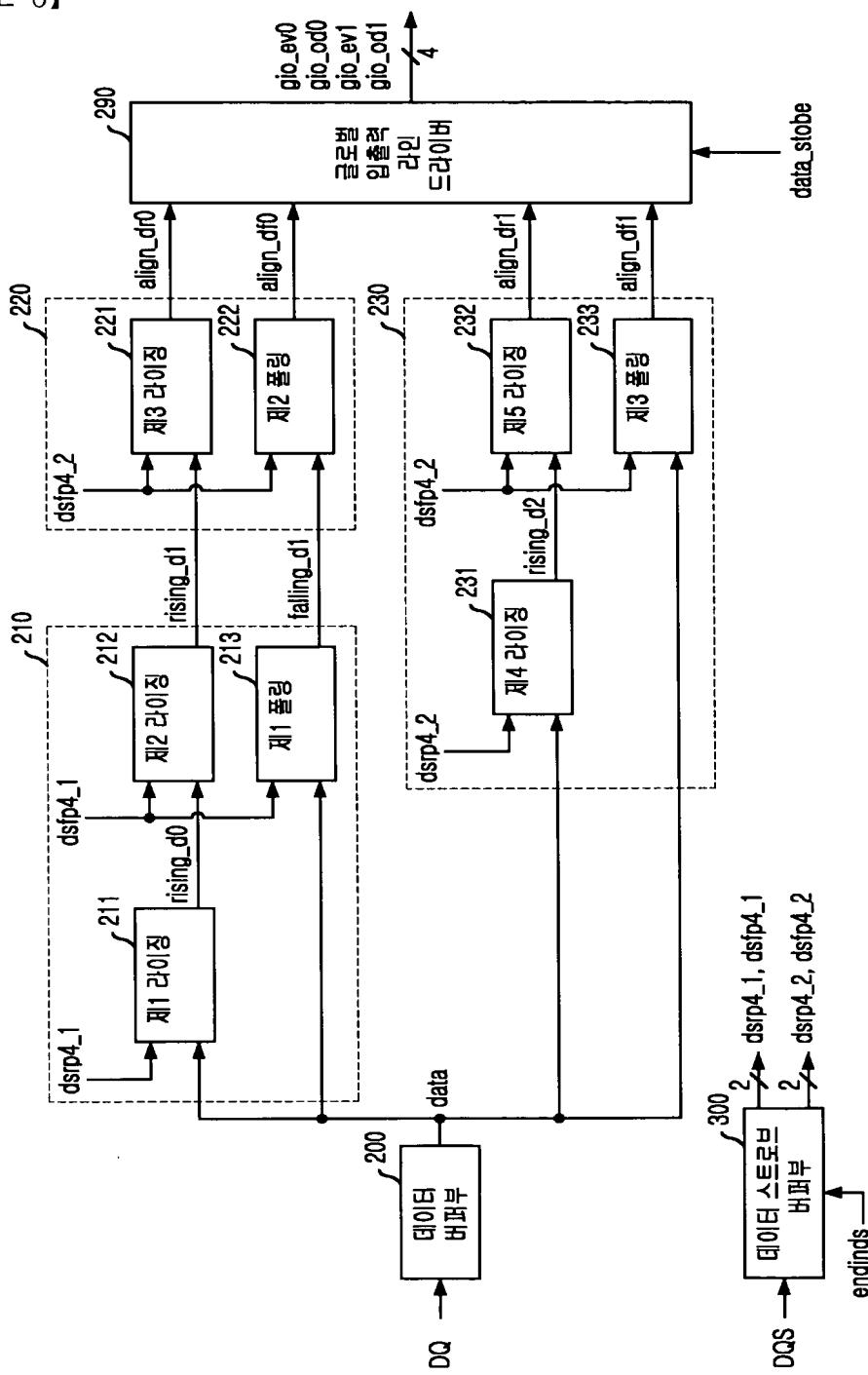
【도 4】



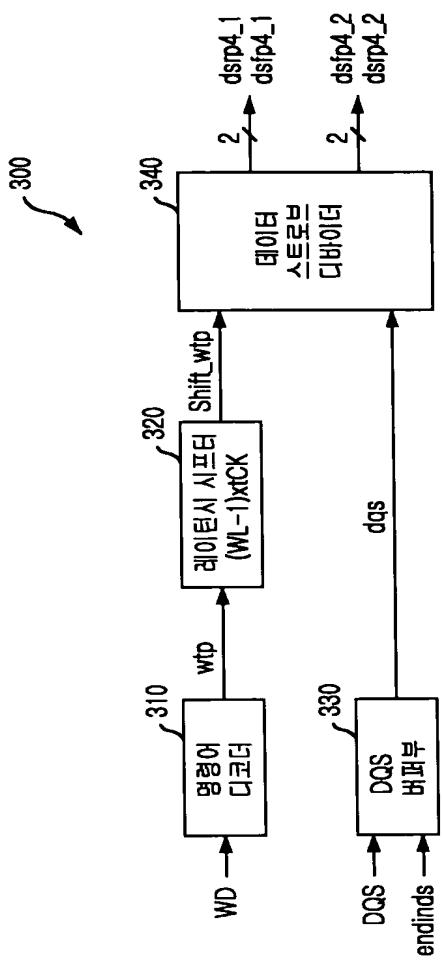
【도 5】



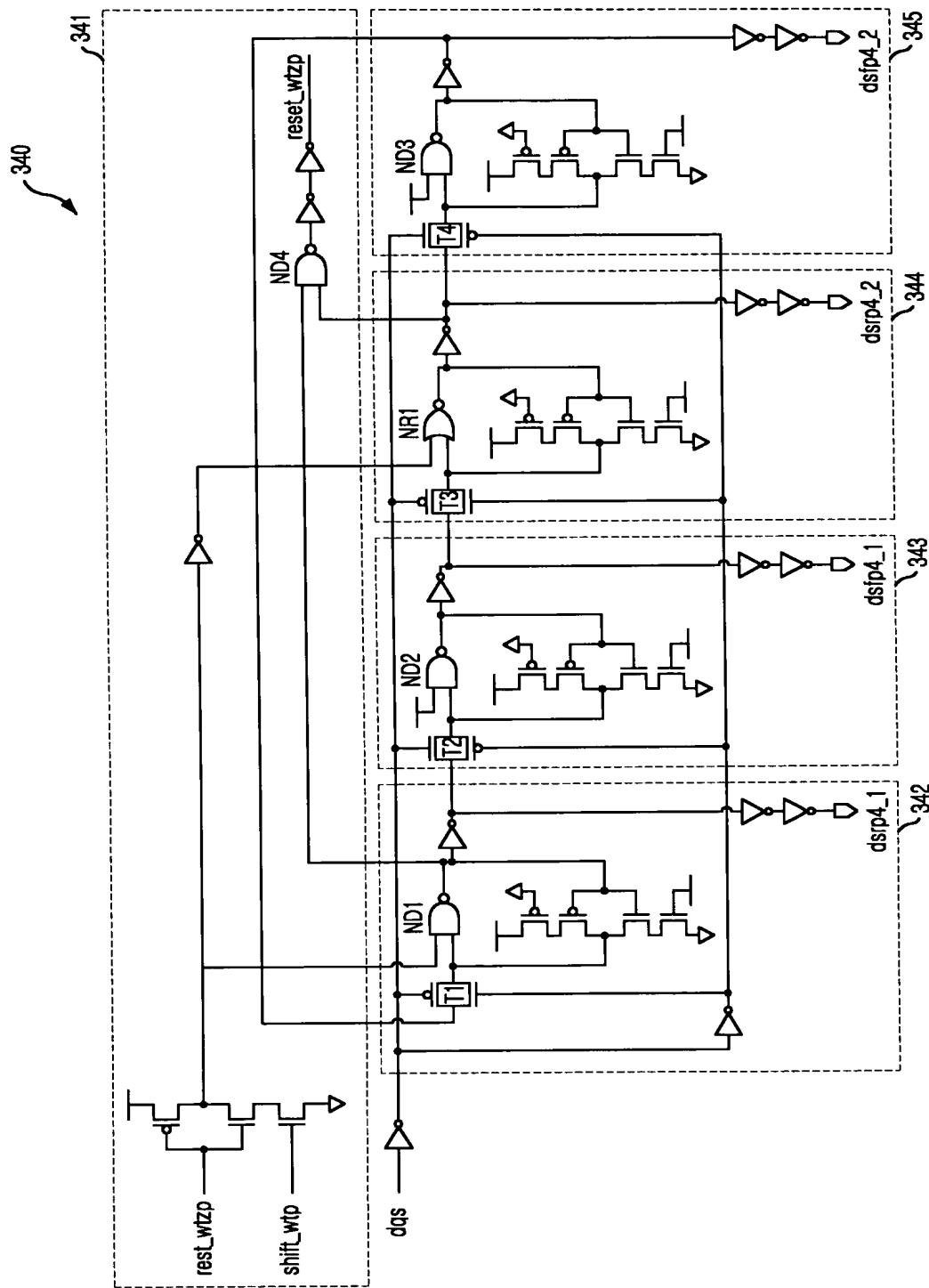
【도 6】



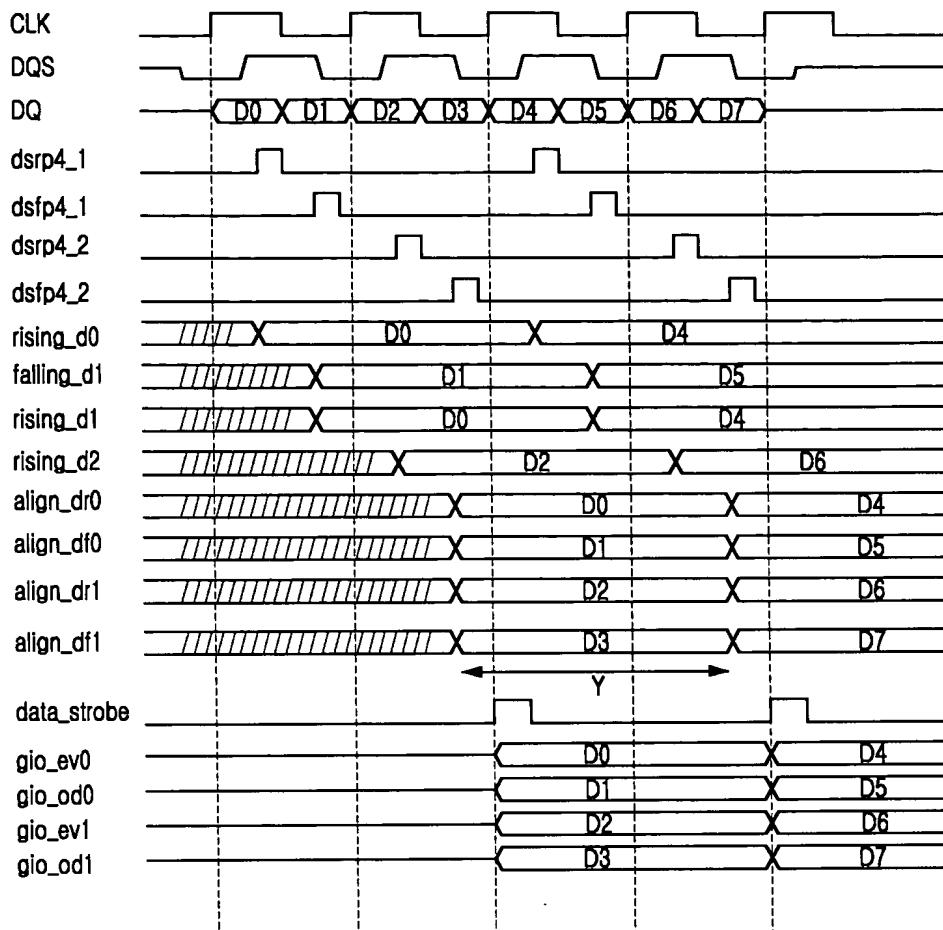
【도 7】



【H 8】



【도 9】



【도 10】

